Internet-based teleoperation: A case study - toward delay approximation and speed limit module

Zhong Shengtong, Philippe Le Parc, Jean Vareille

INTERNET-BASED TELEOPERATION: A CASE STUDY
Toward Delay Approximation and Speed Limit Module

Shengtong Zhong,
Department of Computer Science, Högskolan Dalarna, Röda vägen 3, 78188 Borlänge, Sweden
{h05shezh@du.se}

Philippe Le Parc, Jean Vareille

Laboratoire Informatique des Systèmes Complexes (LISyC), Université de Bretagne Occidentale
20, avenue Le Gorgeu, 29285 Brest Cedex, France
{ philippe.le-parc—jean.vareille}@univ-brest.fr

Keywords: remote control, teleoperation, mobile robot, Internet delay, path error, Delay Approximator, speed limit

Abstract: This paper presents the internet-based remote control of mobile robot. To face unpredictable Internet delays and possible connection rupture, a direct teleoperation architecture with "Speed Limit Module" (SLM) and "Delay Approximator" (DA) is proposed. This direct control architecture guarantees the path error of the robot motion is restricted within the path error tolerance of the application. Experiment results show the effectiveness and applicability of this direct internet control architecture in the real internet environment.

1 INTRODUCTION

Internet is not only an information highway, but also a mean to remotely control mechanical systems, such as robotic devices. But Internet doesn't provide a guaranteed Quality of Service (QoS); it entitles a number of limitation and difficulties, such as bandwidth constraint, transmission delays, packet lost, connection rupture etc. The situation above influence the performance of Internet based telerobotics systems, which is a new field in the recent decade.

The Mercury project (Goldberg et al., 1994) is one of the earliest telerobotics implementation over the internet, then coming with the Telerobot in Australia (Taylor and Dalton, 1997), the painting PumaPaint Robot (Stein, 1998), and Khepera robot (Saucy and Mondada, 2000) etc. During the past ten years, lots of such systems have been introduced by different researchers all over the world.

Most of these researches use a supervisory control scheme which enables operator to issue high level commands. As the internet time delay is unpredictable, the design of direct control scheme which enables user to control the motion of robot continuously may not be easy. Such control schemes have been proposed, but are not adequate to alleviate the influence of Internet time delay.

This paper presents direct teleoperation architecture of a continuous robot motion control which meets path error tolerance under the unpredictable Internet time delay. Here, the path error is guaranteed only if the path error at every turning/stop point is restricted within a path error tolerance which depends on application itself.

Two kinds of control strategy are introduced in the Section 2, followed by the detail of SLM with the quality level idea of GEMMA-Q (Ogor, 2001) and how it works together with the DA to meet the application requirements. During the Section 3, the software implementation is presented with an application over Miabot (two-wheeled robot widely used in soccer competition organized by FIRA).

2 TELEOPERATION ARCHITECTURE

2.1 Generic Architecture

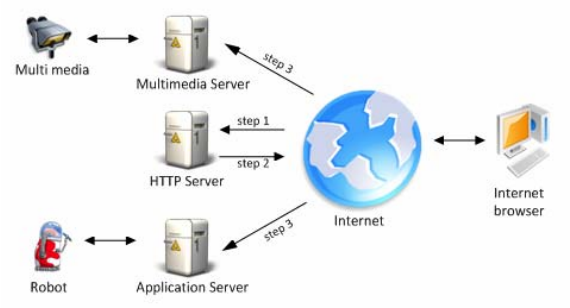

Figure 1: Generic architecture.

The teleoperation over Internet lies on a generic teleoperation architecture (Figure 1), through which commands are exchanged between remote system and operator (user). The user, through his Internet navigator, addresses a request to a Web Server (step1) and downloads an application, such as a Java applet (step 2), on his own workstation. A connection is then established towards the Application Server in charge of robots and client management (step 3). In the same time, another connection with Multimedia Server is also established in the form of exchanging media signals. The user is now able to control the remote robot upon request.

The generic architecture is the same in most applications. The key problem is to alleviate the influence of Internet time delay and towards a continuous control within the path error requirement.

In the following parts, the paper discusses two kinds of control strategies and addresses their advantages and disadvantages. To face the unpredictable Internet time delay, improved direct teleoperation architecture with SLM and DA is proposed.

2.2 Control Strategies

2.2.1 "Move and Wait" Strategy

The " Move and Wait" strategy (Sheridan, 1992), which is typical for space robots with long distance communication has been applied first. In the Miabot case, five commands: "move forward" (MF), "move backward" (MB), "turn left" (TL), "turn right" (TR) and "stop" (ST) have been defined. These commands are enough to perform any complex task, but the user has to send lots of commands and change the move or turn parameters from time to time to meet the requirement.

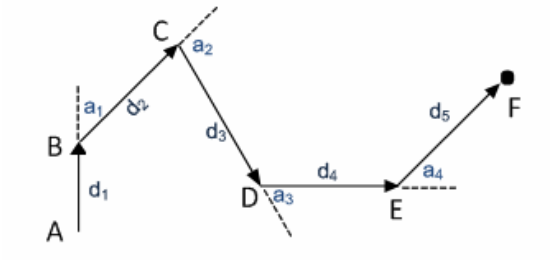

Figure 2: A serial movement task.

In Figure 2, the robot is moving from $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$. This complex task requires the following commands with respective parameters.

Table 1: Sub tasks with "Move and Wait" strategy.

Sub task	Parameter	Location
MF	Distance d_1	$A \rightarrow B$
TR	Degree a_1	B
MF	Distance d_2	$B \rightarrow C$
TR	Degree a_2	C
MF	Distance d_3	$C \rightarrow D$
TL	Degree a_3	D
MF	Distance d_4	$D \rightarrow E$
TL	Degree a_4	E
MF	Distance d_5	$E \rightarrow F$

The user needs eighteen operations, including change parameter (distance and degree) and send command, to achieve this task. The execution result (path) is accurate if each sub task is exactly performed. Due to the physical mechanism of the robot (motor, gear, wheel, etc.), it is possible to have path error, but this situation is not concerned in this paper. Massive operations by the user is the basis of this strategy, which is insensitive to Internet time delay as the user has to wait the completely execution of previous command. The task process is not fluent; the execution time of entire task is much longer than the sum of each sub task time.

2.2.2 "Speed Control" Strategy

To reduce user's workload and towards a fluent process during the complex task, a "Speed control" strategy, similar to real driving, is introduced. "No-stop move forward" (NSMF), "no-stop move backward" (NSMB), "turn left" (TL), "turn right" (TR) and "stop" (ST) commands are defined in this strategy. When the robot executes a turning command, it stops running first, perform the entire

turning next, and then runs in previous speed again. The user may choose different running speed as well as turning degree. Required commands to accomplish the task from Figure 2 are shown below.

Table 2: Sub tasks with "Speed Control" strategy.

Sub task	Parameter	Location
NSMF	Speed v	A $\rightarrow$ B
TR	Degree a_1	B
		B $\rightarrow$ C
TR	Degree a_2	C
		C $\rightarrow$ D
TL	Degree a_3	D
		D $\rightarrow$ E
TL	Degree a_4	E
		E $\rightarrow$ F
ST		F

The significant reduce of the user's operation is observed. The robot is running with speed v during the whole task even though there is a turning/stop point. When the user sends a turning/stop command to robot, the command reaches the robot with a discrete time delay Δt due to Internet time delay. The robot may run an extra distance Δd before it performs the turning/stop.

$$\Delta d = v \times \Delta t$$

Where

v indicates the current speed of the robot, Δt is the current Internet time delay between sending the command from the user and executing the command by robot.

As Internet doesn't provide a guaranteed Quality of Service (QoS), the time delay is unpredictable. Figure 3 is an execution result to the task of Figure 2 with "Speed control" strategy. The path error is accumulative, and path deviation is significant which may result in the failure of the task.

Facing the discrete Internet time delay, to minimize the path deviation (error) and toward a fluent and successful task implementation with "Speed control" strategy, the quality level idea of GEMMA-Q (Quality of Service GEMMA) which derived from GEMMA (ADEPA, 1982) a widely used tool in French industrial, is introduced. Next section describes the SLM with quality level and how this works together with DA to alleviate the influence of discrete Internet time delay. It's functionary to change the robot speed automatically to restrict the path error within path error tolerance δ at every turning/stop point.

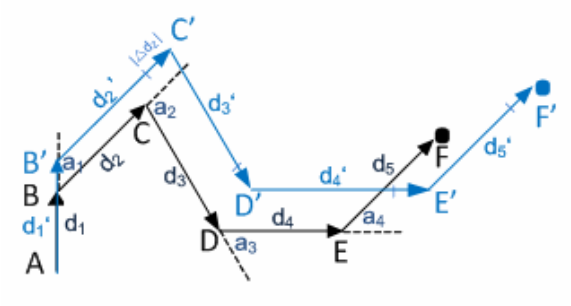

Figure 3: A path result with "Speed Control"

2.3 Speed Limit Module and Delay Approximator

2.3.1 GEMMA-Q approach

The GEMMA methodology is developed to model the different phase of an automated system. It is a base for programming canvas proposing to the integration of a new automated system in the teleoperation architecture. Unfortunately, the GEMMA is only subjected to evolutions on automated system itself or the action undertaken by user owing the system. It doesn't consider the event on the network communication quality.

The GEMMA-Q methodology derived from GEMMA has been proposed. The basic idea is to define the quality of communication as a discrete value. Following the value, the automated system has to follow different rules to ensure the security (path error). According to the type of quality, these rules might be different. In case of a good quality, the automated system must evolutes freely and no constraint is present. If the user has a slightly deteriorated communication quality, the user might keep the control and the automated system has to work in a deteriorated mode where, for example, the movement speed may be slowed down.

2.3.2 Speed Limit Module (SLM)

The GEMMA-Q has been used in the application over robot arm and prototype machine (Le Parc et al., 2001). In this paper, which aims to control a simple robot, use only the quality level idea of GEMMA-Q to design SLM. n communication quality levels are introduced which corresponds Internet time delay

zone (Time D.). Each time delay zone has its own sub path error tolerance $\Delta\varepsilon_j$ (Sub T.) and Max Speed v_j as shown in the Table 3.

Table 3: Quality levels in SLM

Qual.	Time D.	Sub T.	Max Speed
Q_0	$0 \sim t_0$	$\Delta\delta_0$	$v_0 = \Delta\delta_0 / t_0$
Q_1	$t_0 \sim t_1$	$\Delta\delta_1$	$v_1 = \Delta\delta_1 /(t_1 - t_0)$
Q_2	$t_1 \sim t_2$	$\Delta\delta_2$	$v_2 = \Delta\delta_2 /(t_2 - t_1)$
Q_3	$t_2 \sim t_3$	$\Delta\delta_3$	$v_3 = \Delta\delta_3 /(t_3 - t_2)$
...	...	...	...
Q_{n-1}	$t_{n-1} \sim t_n$	$\Delta\delta_{n-1}$	$v_{n-1} = \Delta\delta_{n-1} /(t_n - t_{n-1})$
Q_n	$\geq t_n$	0	$v_n = 0$

When the quality level changes, the Application Server evaluates the Current Robot Speed (CRS) and the Max Speed of this quality level (MSoT). If CRS $\leq$ MSoT, no command is sent to robot; else the Application Server sends MSoT command to robot (change CRS to MSoT).

MSoT is calculated by the time delay zone and sub tolerance which are defined by user according to application. In order to meet the path error tolerance δ at the turning/stop point, the following constraints are used when designing SLM:

$$\begin{cases} \sum_{j=0}^{n} \Delta\delta_j \leq \delta \\ v_{j+1} \leq v_j \quad j \in (0,1,\cdots n-1) \end{cases}$$

The above constraint guarantees the path error at the turning/stop point is restricted within δ in any Internet time delay situation. Q_0 is the best quality level, its Max Speed v_0 is the fastest running speed of the robot. Q_n is the disconnection situation, and the robot stops running immediately $(v_n = 0)$. The proof of how the constraint works are shown below.

There are two kinds of situation when the robot runs under the speed limit rules.

1. Unstable network delay

The robot is running in the different time delay quality levels between two continuous actual Internet time delays. The worst case is from Q_0 to Q_n:

$$\begin{aligned} \Delta d &= \Delta\delta_0 + \Delta\delta_1 + \cdots + \Delta\delta_n \\ &= \sum_{j=0}^{n} \Delta\delta_j \\ &\leq \delta \end{aligned}$$

2. Stable network delay

It means the robot is running in the same time delay quality level Q_i between two continuous actual Internet time delay (from the real clock time of receiving previous actual Internet time delay to the real clock time of receiving next actual Internet time delay). Then

$$\begin{aligned} \Delta d &= v_i \times t_i \\ &\leq v_0 \times t_0 + v_1 \times (t_1 - t_0) + \cdots + v_i (t_i - t_{i-1}) \\ &= \Delta\delta_0 + \Delta\delta_1 + \cdots + \Delta\delta_i \\ &= \sum_{j=0}^{i} \Delta\delta_j \\ &\leq \delta \end{aligned}$$

The path error Δd is within the restriction of path error tolerance δ in both situations. Then the path error at every turning/stop point in the task is guaranteed and the successfully continuous control of the robot is achieved.

With the constraints, the different quality levels with its respective time delay zone value, sub tolerance and max speed are defined according to different application. e.g.: the normal quality level of Internet time delay is Q_j in the application, its sub tolerance $\Delta\delta_j$ should take a larger percentage of δ. It means the robot is preferred to have larger sub tolerance in the normal quality level; in the same time, speed limit rules guarantees $\Delta d \leq \delta$ in any situation.

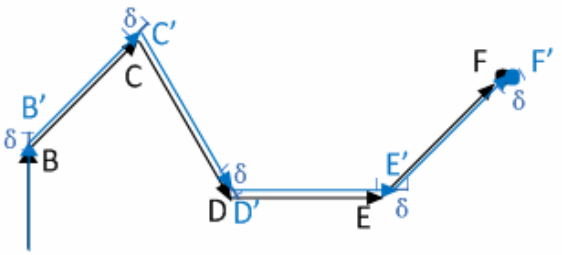

Figure 4: A path result using "Speed Control" with SLM

In Figure 4, the path deviation at every turning/stop point is restricted within δ. Then a successful task is achieved using "Speed control" with SLM.

The above describes the detail SLM with the quality level idea, next part emphasize on how the gets the quality level information of current time delay from DA.

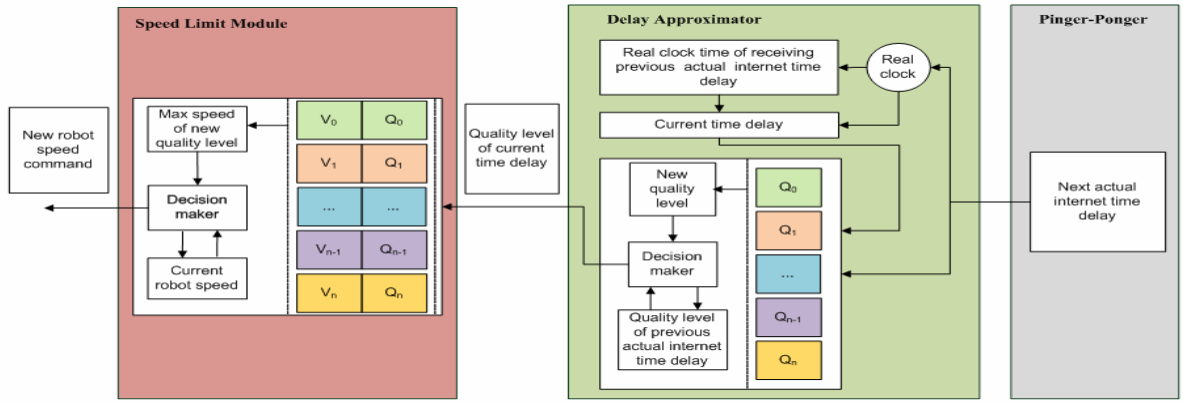

Figure 5: Global view of SLM and DA

2.3.3 Delay Approximator (DA) principals

In the generic architecture, Pinger-Ponger is the mechanism to provide the Internet time delay information to DA. The next actual Internet time delay Δt_n is calculated as follows:

$$\Delta t_n = T_n - T_p$$

Where

T_p is the real clock time of receiving previous actual Internet time delay from Pinger-Ponger; T_n is the real clock time when Pinger-Ponger sends the next Internet time delay to DA. With above feature, the current (up-to-now) time delay Δt_c from T_p is formed as:

$$\begin{cases} \Delta t_c = t_j & j = 0 \\ \Delta t_c = t_j - t_{j-1} & j \in (1,2,\cdots n) \end{cases}$$

Where

t_j is the time delay zone in SLM. Δt_c is the watchdog and j is the automatic counter initialized with 0. Δt_c is only activated when there is quality level change, and this quality level change is used as current time delay change information.

Pinger-Ponger only provides the Internet time delay when it gets one. The Internet time delay is unpredictable, so there is no idea when Pinger-Ponger gets new information.

There is no idea about the future, but the current situation is supervised by DA as following principals:

1. DA receives information from Pinger-Ponger.

Pinger-Ponger informs DA of receiving the next actual Internet time delay, and then DA forwards the quality level of calculated Δt_n to SLM. Meanwhile the previous parameters are set to be the current ones: the value of T_p is set to T_n, the quality level of previous actual Internet time is changed to the quality level of Δt_n, reset the watchdog (Δt_c).

2. No information from Pinger-Ponger and Δt_c is activated to a quality level change.

Δt_p is the previous actual Internet time delay. There are two kinds of situations:

- $\Delta t_c \le \Delta t_p$

  The quality level of current time delay is no worse than that of previous actual time delay; there is no action and DA keeps supervising.

- $\Delta t_c > \Delta t_p$

  The quality level of current time delay situation is worse than that of previous actual Internet time delay. When Δt_c is activated, it indicates a change of quality level and "Delay Approximator" forwards the new quality level to SLM. DA keeps supervising.

DA keeps supervising the Internet delay situation all the time. It provides the real Internet time delay or current time delay to "Speed Limit Schema".

2.3.4 Global view

In figure 5, SLM receives quality level information of current time delay which is decided by DA, and then "Speed Limit Schema" applies the speed limit rules accordingly. This guarantees the robot is running in a proper speed that meets the restriction of path error tolerance at every turning/stop point. Then a continuous control of robot motion with "Speed Control" strategy is achieved.

Figure 6: Experimental senario

3 EXPERIMENT

The remote control system server has been built in UBO, France. Figure 6 is a screenshot of the user interface and the left side of the picture is the live camera of the mobile robot. Some users (mainly from France, Sweden, and China) are invited to test the feasibility of effectiveness of the system. The average Internet time delay varies a lot between European users and Asian users. It's not easy to find a uniform SLM for all the users, but certain SLM works fine for some users. The test is still on-going currently. The average Internet time delay of the user is able to observe during the test, so next step consideration is to build SLM for respective user group (distinguished by average Internet time delay. e.g.: Asia, Europe, etc.) and the system can choose different SLM automatically due to the user group.

4 CONCLUSION

In this paper, the SLM and DA based architecture is proposed to face the unpredictable Internet delay in Internet-based robot control. This approach guarantees the path error of the continuous robot motion. Here, the path error is guaranteed only if the path error at every turning/stop point is within a path error tolerance δ which depends on application itself.

In this architecture, the current time delay is supervised by DA; and SLM applies different speed limit rules according to the current time delay situation. Then the robot is always running in a proper speed which meets the path error restriction. Finally, a continuous control of the internet-based robot is achieved successfully.

REFERENCES

Andreu, D., Fraisse, P., Roqueta, V., Zapata, R., 2003. Internet enhanced teleoperation toward a remote supervised delay regulator. ICIT'03, *IEEE Int. Conf. on Industrial Technology, Maribor.*

Ogor, P., 2001. Une architecture générique pour la supervision sure à distance de machines de production avec Internet. *Ph.D. thesis, Université de Bretagne Occidentale.*

ADEPA, 1981. GEMMA: Guide d'études des modes de marche et d'arrêt.

Meng Wang, and James N.K. Liu, 2005. Interactive control for Internet-based mobile robot teleoperation. *Robotics and Autonomous Systems, Volume 52, Issues 2-3, 31 August 2005, Pages 160-179.*

K.-H. Han, J.-H. Kim, 2002. Direct Internet Control Architecture for Personal Robot. *Proc. of the 2002 FIRA Robot World Congress, pp. 264-268, Seoul.*

P. Ogor, P. Le Parc, J. Vareille et L. Marcé, 2001. Control a Robot on Internet. *6th IFAC Symposium on Cost Oriented Automation, Berlin.*

J. Vareille, P. Le Parc et L. Marcé, 2004. Web remote control of mechanical systems: delay problems and experimental measurements of Round Trip Time. *2nd Workshop CNRS-NSF Applications of Time-Delay systems, Nantes.*